EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L14	159	714/6.ccls. and @pd>="20070520"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/09 14:41
L15	30	714/36.ccls. and @pd>="20070520"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/09 13:45
L16	578	"shadow memory"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/09 15:16
L17	127	"shadow memory" and (crc or ecc or "verification data" or "redundancy data")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/09 13:46
L18	755	714/7.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/09 14:42
L19	9	714/7.ccls. and "shadow memory"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/09 14:42
L20	1144	714/5.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/09 14:44
L21	3	714/5.ccls. and "shadow memory"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/09 14:43

EAST Search History

L22	840	714/42.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/09 14:43
L23	7	714/42.ccls. and "shadow memory"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/09 14:44
L24	285	714/52.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/09 14:44
L25	2	714/52.ccls. and "shadow memory"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/09 15:14
S1	56	(shadow ADJ memory).TI.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 09:12
S2	0	10/731661	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 08:43
S3	0	(shadow ADJ memory) WITH (verif$3 OR verification) WITH (compar$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 09:12
S4	2	(shadow ADJ memory) WITH (verif$3 OR verification)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 09:13

EAST Search History

S5	16	(shadow ADJ memory) WITH (compar$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 09:13
S6	493	shadow ADJ memory	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 09:41
S7	36	S6 WITH test$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 09:29
S8	178	S6 AND (boot$4 OR BIOS)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 09:39
S9	72	S6 SAME (boot$4 OR BIOS)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 09:46
S10	848	shadow ADJ (memory OR RAM)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 09:42
S11	232	714/36.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/20 09:20
S12	0	S10 AND S11	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 09:42

EAST Search History

S13	5	S11 AND shadow	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 09:42
S14	72	S9 SAME (boot$4 OR BIOS)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 10:31
S15	208	S10 SAME (boot$4 OR BIOS)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 09:46
S16	25	S15 SAME (verif$3 OR verification OR compar$4)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 09:47
S17	103	S15 SAME (verif$3 OR verification OR compar$4 OR test$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 10:12
S18	80	shadow WITH "EEPROM"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 09:51
S19	13	S18 SAME (verif$3 OR verification OR compar$4 OR test$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 10:55
S20	6	(backup NEAR updat$3).TI. AND memor$3.AB.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 10:20

EAST Search History

S21	128	(backup NEAR updat$3).AB.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 10:17
S22	8	(backup NEAR updat$3) WITH (compar$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 10:18
S23	50	BIOS NEAR backup	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 10:19
S24	162	(boot$4 OR BIOS) NEAR backup	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 10:20
S25	17	S24 SAME (verif$3 OR verification OR compar$4 OR test$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 10:20
S26	16	S24 WITH updat$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 10:20
S27	819	(boot$4 OR BIOS) WITH backup	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 10:31
S28	50	S27 WITH (updat$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 10:32

EAST Search History

S29	50	S28 NOT S6	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 10:42
S30	303	mirror NEAR updat$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 10:43
S31	33	S30 SAME memory	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 10:54
S32	53765	"714"/$.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 10:54
S33	712	S32 AND shadow	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 10:55
S34	292	S32 AND shadow SAME ((verif$3 OR verification OR compar$4 OR test$3))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 10:55
S35	171	S32 AND shadow WITH ((verif$3 OR verification OR compar$4 OR test$3))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 11:54
S36	403	EEPROM WITH copy	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 11:54

EAST Search History

S37	682	EEPROM WITH (copy$3 OR copie$1)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 11:54
S38	42	S37 SAME compar$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 13:52
S39	1758	714/6.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 13:52
S40	147	S39 AND updat$3 WITH mirror$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 13:54
S41	13	S39 AND lockstep	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 14:04
S42	109	S39 AND coherenc$2	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 14:04
S43	12	S39 AND coherenc$2.TI.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 14:25
S44	963	S39 AND compar$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 14:25

EAST Search History

S45	316	S44 AND (timeout OR periodic OR periodically OR countdown)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 14:31
S46	30	S39 AND (compar$4 WITH clock$1)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/09 14:31
S47	25	("4979108" | "5138710" | "5257391" | "5261058" | "5280487" | "5379417" | "5388108" | "5418925" | "5471640" | "5487160" | "5542065" | "5546558" | "5548712" | "5553230" | "5778206" | "5799022").PN. OR ("5953352"). URPN.	US-PGPUB; USPAT; USOCR	OR	ON	2006/05/09 16:10
S49	1107	RAID ADJ "1"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/10 08:05
S50	51	S49 AND (((power OR turn) ADJ on) OR (power ADJ up))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/10 08:15
S51	94945	EEPROM	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/10 08:16
S52	220	EEPROM NEAR test$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/10 08:28
S53	12	(RAM NEAR test$3) WITH EEPROM	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/10 08:30

EAST Search History

S54	30	(RAM NEAR (compar$3 OR match$3)) WITH EEPROM	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/10 08:30
S55	25	timer WITH EEPROM WITH test$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/10 09:26
S56	39	timer WITH EEPROM WITH compar$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/10 09:29
S57	39	S56 NOT S55	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/10 09:26
S58	11	interval WITH EEPROM WITH compar$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/10 09:31
S59	29775	RAM WITH EEPROM	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/10 09:31
S60	14	(RAM WITH EEPROM) SAME (compar$3 WITH (interval OR clock))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/10 09:32
S61	103	(EEPROM) SAME (compar$3 WITH (interval OR clock))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/10 09:32

EAST Search History

S63	89	S61 NOT S60	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/10 09:33
S64	60	714/36.ccls. and @pd>="20060510"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/20 09:40
S65	447	714/6.ccls. and @pd>="20060510"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/11/09 13:30
S66	25	714/6.ccls. and @pd>="20060510" and shadow	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/20 09:45
S67	930	"shadow memory" or "shadow ram"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/20 09:45

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L26	73	714/36.ccls.	US-PGPUB	OR	ON	2007/11/09 15:14
L27	185	"shadow memory"	US-PGPUB	OR	ON	2007/11/09 15:16

Web Results **1 - 10** of about **75,600** for **shadow memory verification data**. (**0.16** seconds)

Tamper resistant **shadow memory** - Patent 7188282
Tamper resistant **shadow memory**. Abstract. An integrated circuit comprising a processor ..... 371 shows an input signature **verification data** format for Test ...
www.patentmonkey.com/PM/patentid/7188282.aspx - 138k - Cached - Similar pages

High speed testing for programmable logic devices - Patent 5159599
a shift register connected to said **memory** array for writing **data** onto the .... Test logic 22 is connected to the **shadow** RAM 20 and the shift register 16, ...
www.freepatentsonline.com/5159599.html - 32k - Cached - Similar pages

Methods and apparatus for verifying the operation of a circuit ...
as said **data** is read from said at least one buffer, writing said **data** into said RAM. 2. The method of claim 1 wherein the functional **verification** system ...
www.freepatentsonline.com/EP1441296.html - 16k - Cached - Similar pages

DRAM **memory** cell for programmable logic devices - US Patent 5847577
The circuit of claim 1 further comprising a **shadow memory** array comprising .... Bradley Felton and Neil Hastie, "2.6 Configuration **Data Verification** and the ...
www.patentstorm.us/patents/5847577-claims.html - 24k - Cached - Similar pages

Automated safestore stack generation and recovery in a fault ...
C) shadowing, packing and validity **verification** means coupled ... of activity means further includes a cache **memory**, the contents of said **shadow** set being ...
www.patentstorm.us/patents/5557737-claims.html - 17k - Cached - Similar pages
[More results from www.patentstorm.us]

[PPT] Formal **Verification** of Pipelined Processors
File Format: Microsoft Powerpoint - View as HTML
Description of **Verification**. – 21 –. Auxiliary **Data** Structures. **Shadow** Fields ... **Shadow Memory** Address Map. Latest non-speculative instruction to modify a ...
www.cs.cmu.edu/~bryant/presentations/cav03-ooo.ppt - Similar pages

[PDF] **Verification** of Pipelined Microprocessors by Comparing **Memory** ...
File Format: PDF/Adobe Acrobat - View as HTML
memory writes. However, we need bit-level **data** for various **memory** locations in order ..... **memory** element in it with empty original and **shadow memory** lists. ...
www.cs.cmu.edu/~bryant/pubdir/asian97.pdf - Similar pages
[More results from www.cs.cmu.edu]

new **memory** not regnised ...
Generic volume **shadow** copy USB 2 HS-CF Virus Protection [Back to Top] ... no **verification data** KB927978 on 19/06/2007 (details...) ...
forums.vnunet.com/message.jspa?messageID=944944 - 54k - Cached - Similar pages

Bios error codes and numbers
51 **Memory** size adjusted due to relocation/ **shadow**. **Memory** test above 1M to follow. ...
61 Display **memory verification** over. About to go for DMA #1 base ...
www.geocities.com/budallen98_98/bios_error_codes.html - 23k - Cached - Similar pages

CommsDesign - Challenges in HyperTransport **Verification**
As HyperTransport designs proliferate the comm sector, **verification** will become a ... and performs automatic read and write checking via a **shadow memory**. ...
www.commsdesign.com/design_corner/showArticle.jhtml?articleID=16505887 - 41k - Cached - Similar pages

1 2 3 4 5 6 7 8 9 10 **Next**

Terms used: **shadow memory verification data** Found **69,133** of **214,158**

Sort results by relevance
Display results expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 Functional verification of ESL models: Memory modeling in ESL-RTL equivalence checking
Alfred Koelbl, Jerry R. Burch, Carl Pixley
June 2007 **Proceedings of the 44th annual conference on Design automation DAC '07**
Publisher: ACM Press
Full text available: pdf(131.55 KB) Additional Information: full citation, abstract, references, index terms

When designers create RTL models from a system-level specification, arrays in the system-level model are often implemented as memories in the RTL. Knowing the correspondence between ESL arrays and RTL memories can significantly reduce the complexity of a formal equivalence check between the ESL model and the RTL. In practice, however, handling memory mappings in ESL-RTL equivalence checking is non-trivial for the following reasons: First, because of a lack of bit-accurate data-types in the sy ...

Keywords: ESL, equivalence checking, memory, verification

2 Automated analysis: Control-flow integrity
Martín Abadi, Mihai Budiu, Úlfar Erlingsson, Jay Ligatti
November 2005 **Proceedings of the 12th ACM conference on Computer and communications security CCS '05**
Publisher: ACM Press
Full text available: pdf(218.60 KB) Additional Information: full citation, abstract, references, citings, index terms

Current software attacks often build on exploits that subvert machine-code execution. The enforcement of a basic safety property, Control-Flow Integrity (CFI), can prevent such attacks from arbitrarily controlling program behavior. CFI enforcement is simple, and its guarantees can be established formally even with respect to powerful adversaries. Moreover, CFI enforcement is practical: it is compatible with existing software and can be done efficiently using software rewriting in commodity syste ...

Keywords: binary rewriting, control-flow graph, inlined reference monitors, vulnerabilities

3 Functional verification methodology of Chameleon processor

Françoise Casaubieilh, Anthony McIsaac, Mike Benjamin, Mike Bartley, François Pogodalla, Frédéric Rocheteau, Mohamed Belhadj, Jeremy Eggleton, Gérard Mas, Geoff Barrett, Christian Berthet
June 1996 **Proceedings of the 33rd annual conference on Design automation DAC '96**
Publisher: ACM Press
Full text available: pdf(62.38 KB) Additional Information: full citation, references, citings, index terms

4 Cryptography and data security
Dorothy Elizabeth Robling Denning
January 1982 Book
Publisher: Addison-Wesley Longman Publishing Co., Inc.
Full text available: pdf(19.47 MB) Additional Information: full citation, abstract, references, cited by, index terms

From the Preface (See Front Matter for full Preface)

Electronic computers have evolved from exiguous experimental enterprises in the 1940s to prolific practical data processing systems in the 1980s. As we have come to rely on these systems to process and store data, we have also come to wonder about their ability to protect valuable data.

Data security is the science and study of methods of protecting data in computer and communication systems from unauthorized disclosure ...

5 Efficient Field P ocessing Cores in an Innovative Protocol Processo System-on-Chip
G. Lykakis, N. Mouratidis, K. Vlachos, N. Nikolaou, S. Perissakis, G. Sourdis, G. Konstantoulakis, D. Pnevmatikatos, D. Reisis
March 2003 **Proceedings of the conference on Design, Automation and Test in Europe: Designers' Forum - Volume 2 DATE '03**
Publisher: IEEE Computer Society
Full text available: pdf(179.42 KB) Publisher Site Additional Information: full citation, abstract, index terms

We present an innovative protocol processor component that combines wire-speed processing for low-level, and best effort processing for higher-level protocols. The component is a System-on-Chip that integrates variable size packet buffering, specialised cores for header and field processing, generic RISC cores and scheduling blocks. We focus on the main innovation, the reprogrammable pipeline module, and discuss its internal architecture, optimised to perform field processing on byte streams, as ...

6 Functional verification of a multiple-issue, out-of-order, superscalar Alpha processor—the DEC Alpha 21264 microprocessor
Scott Taylor, Michael Quinn, Darren Brown, Nathan Dohm, Scot Hildebrandt, James Huggins, Carl Ramey
May 1998 **Proceedings of the 35th annual conference on Design automation DAC '98**
Publisher: ACM Press
Full text available: pdf(153.68 KB) Additional Information: full citation, abstract, references, citings, index terms

DIGITAL's Alpha 21264 processor is a highly out-of-order, superpipelined, superscalar implementation of the Alpha architecture, capable of a peak execution rate of six instructions per cycle and a sustainable rate of four per cycle. The 21264 also features a 500 MHz clock speed and a high-bandwidth system interface that channels up to 5.3 Gbytes/second of cache data and 2.6 Gbytes/second of main-memory data into the

processor. Simulation-based functional verification was performed on the lo ...

Keywords: 21264, Alpha, architecture, coverage anaysis, microprocessor, pseudo-random, validation, verification

7 Area efficient architectures for information integrity in cache memories

Seongwoo Kim, Arun K. Somani

May 1999 **ACM SIGARCH Computer Architecture News , Proceedings of the 26th annual international symposium on Computer architecture ISCA '99**, Volume 27 Issue 2

Publisher: IEEE Computer Society, ACM Press

Full text available: pdf(227.09 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Information integrity in cache memories is a fundamental requirement for dependable computing. Conventional architectures for enhancing cache reliability using check codes make it difficult to trade between the level of data integrity and the chip area requirement. We focus on transient fault tolerance in primary cache memories and develop new architectural solutions, to maximize fault coverage when the budgeted silicon area is not sufficient for the conventional configuration of an error checki ...

8 Streamlining inter-operation memory communication via data dependence prediction

Andreas Moshovos, Gurindar S. Sohi

December 1997 **Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture MICRO 30**

Publisher: IEEE Computer Society

Full text available: pdf(1.73 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

We revisit memory hierarchy design viewing memory as an inter-operation communication agent. This perspective leads to the development of novel methods of performing inter-operation memory communication. We use data dependence prediction to identify and link dependent loads and stores so that they can communicate speculatively without incurring the overhead of address calculation, disambiguation and data cache access. We also use data dependence prediction to convert, DEF-store-load-USE chains w ...

Keywords: DEF-store-load-USE chains, address calculation, address disambiguation, communication latency, data cache access, data cache bandwidth requirements, data dependence prediction, instruction window, inter-operation memory communication, memory architecture, memory dependences, memory hierarchy design, storage management, transient value cache

9 Operating system security: SecVisor: a tiny hypervisor to provide lifetime kernel code integrity for commodity OSes

Arvind Seshadri, Mark Luk, Ning Qu, Adrian Perrig

October 2007 **Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles SOSP '07**

Publisher: ACM Press

Full text available: pdf(264.11 KB) Additional Information: full citation, abstract, references, index terms

We propose SecVisor, a tiny hypervisor that ensures code integrity for commodity OS kernels. In particular, SecVisor ensures that only user-approved code can execute in kernel mode over the entire system lifetime. This protects the kernel against code injection attacks, such as kernel rootkits. SecVisor can achieve this propertyeven against

an attacker who controls everything but the CPU, the memory controller, and system memory chips. Further, SecVisor can even defend against attackers with ...

Keywords: code attestation, code injection attacks, code integrity, hypervisor, memory virtualization, preventing

10 Slackened Memory Dependence Enforcement: Combining Opportunistic Forwarding with Decoupled Verification

Alok Garg, M. Wasiur Rashid, Michael Huang

May 2006 **ACM SIGARCH Computer Architecture News , Proceedings of the 33rd annual international symposium on Computer Architecture ISCA '06**, Volume 34 Issue 2

Publisher: IEEE Computer Society, ACM Press

Full text available: pdf(348.27 KB) Additional Information: full citation, abstract, index terms

An efficient mechanism to track and enforce memory dependences is crucial to an out-of-order microprocessor. The conventional approach of using cross-checked load queue and store queue, while very effective in earlier processor incarnations, suffers from scalability problems in modern high-frequency designs that rely on buffering many in-flight instructions to exploit instruction-level parallelism. In this paper, we make a case for a very different approach to dynamic memory disambiguation. We m ...

11 Decoupled hardware support for distributed shared memory

Steven K. Reinhardt, Robert W. Pfile, David A. Wood

May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture ISCA '96**, Volume 24 Issue 2

Publisher: ACM Press

Full text available: pdf(1.47 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper investigates hardware support for fine-grain distributed shared memory (DSM) in networks of workstations. To reduce design time and implementation cost relative to dedicated DSM systems, we decouple the functional hardware components of DSM support, allowing greater use of off-the-shelf devices.We present two decoupled systems, Typhoon-0 and Typhoon-1. Typhoon-0 uses an off-the-shelf protocol processor and network interface; a custom access control device is the only DSM-specific hard ...

12 Session summaries from the 17th symposium on operating systems principle (SOSP'99)

Jay Lepreau, Eric Eide

April 2000 **ACM SIGOPS Operating Systems Review**, Volume 34 Issue 2

Publisher: ACM Press

Full text available: pdf(3.15 MB) Additional Information: full citation, index terms

13 Fast and accurate hierarchical radiosity using global visibility

Frédo Durand, George Drettakis, Claude Puech

April 1999 **ACM Transactions on Graphics (TOG)**, Volume 18 Issue 2

Publisher: ACM Press

Full text available: pdf(8.48 MB) Additional Information: full citation, abstract, references, citings, index terms

Recent hierarchical global illumination algorithms permit the generation of images with a high degree of realism. Nonetheless, appropriate refinement of light transfers, high

quality meshing, and accurate visibility calculation can be challenging tasks. This is particularly true for scenes containing multiple light sources and scenes lit mainly by indirect light. We present solutions to these problems by extending a global visibility data structure, the Visibility Skeleton. This extension ...

Keywords: discontinuity meshing, form factor calculation, global illumination, global visibility, hierarchical radiosity, hierarchical triangulation, perception

14 Model and verification of a data manager based on ARIES

Dean Kuo
December 1996 **ACM Transactions on Database Systems (TODS)**, Volume 21 Issue 4
Publisher: ACM Press

Full text available: pdf(813.93 KB) Additional Information: full citation, abstract, references, citings, index terms, review

In this article, we model and verify a data manager whose algorithm is based on ARIES. The work uses the I/O automata method as the formal model and the definition of correctness is defined on the interface between the scheduler and the data manager.

Keywords: ARIES, I/O automata, system failures

15 Dynamic Verification of Sequential Consistency

Albert Meixner, Daniel J. Sorin
May 2005 **ACM SIGARCH Computer Architecture News , Proceedings of the 32nd annual international symposium on Computer Architecture ISCA '05**, Volume 33 Issue 2
Publisher: IEEE Computer Society, ACM Press
Full text available: pdf(146.68 KB) Additional Information: full citation, abstract, index terms

In this paper, we develop the first feasibly implementable scheme for end-to-end dynamic verification of multithreaded memory systems. For multithreaded (including multiprocessor) memory systems, end-to-end correctness is defined by its memory consistency model. One such consistency model is sequential consistency (SC), which specifies that all loads and stores appear to execute in a total order that respects program order for each thread. Our design, DVSC-Indirect, performs dynamic verification ...

16 Design space optimization of embedded memory systems via data remapping

Krishna V. Palem, Rodric M. Rabbah, Vincent J. Mooney, Pinar Korkmaz, Kiran Puttaswamy
June 2002 **ACM SIGPLAN Notices , Proceedings of the joint conference on Languages, compilers and tools for embedded systems: software and compilers for embedded systems LCTES/SCOPES '02**, Volume 37 Issue 7
Publisher: ACM Press

Full text available: pdf(382.47 KB) Additional Information: full citation, abstract, references, citings, index terms

In this paper, we provide a novel compile-time *data remapping* algorithm that runs in linear time. This remapping algorithm is the first fully automatic approach applicable to pointer-intensive dynamic applications. We show that data remapping can be used to significantly reduce the *energy consumed* as well as the *memory size* needed to meet a user-specified performance goal (i.e., execution time) -- relative to the same application executing without being remapped. These twin ...

Keywords: data remapping, design space exploration, power aware

17 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research CASCON '97**

Publisher: IBM Press

Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

18 Distributed environment: Analysis of X.500 distributed directory refresh strategies

David W. Bachmann, Michael A. Bauer, J. Michael Bennett, Guy A. Fasulo, Michael H. Kamlet, Kevin H. Klinge, Sailesh Makkapati, Jacob Slonim, Toby J. Teorey

October 1991 **Proceedings of the 1991 conference of the Centre for Advanced Studies on Collaborative research CASCON '91**

Publisher: IBM Press

Full text available: pdf(1.12 MB) Additional Information: full citation, abstract, references, citings

Refresh strategies for distributed database directories, commonly recommended for the X.500 standard, are defined and analytically modeled for variations on push/pull and total/differential options under ideal asynchronous control conditions. The models are implemented in a HyperCard-based tool called DirMod ("Directory Model"). Experimental test results show an important elapsed time/performance trade-off among the different strategies, and test data contribute to the verification of the models ...

19 On the power of the frame buffer

Alain Fournier, Donald Fussell

April 1988 **ACM Transactions on Graphics (TOG)**, Volume 7 Issue 2

Publisher: ACM Press

Full text available: pdf(1.95 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Raster graphics displays are almost always refreshed out of a frame buffer in which a digital representation of the currently visible image is kept. The availability of the frame buffer as a two-dimensional memory array representing the displayable area in a screen coordinate system has motivated the development of algorithms that take advantage of this memory for more than just picture storage. The classic example of such an algorithm is the depth buffer algorithm for determining visible s ...

20 Efficient and flexible architectural support for dynamic monitoring

Yuanyuan Zhou, Pin Zhou, Feng Qin, Wei Liu, Josep Torrellas

March 2005 **ACM Transactions on Architecture and Code Optimization (TACO)**, Volume 2 Issue 1

Publisher: ACM Press

Full text available: pdf(524.21 KB) Additional Information: full citation, abstract, references, index terms, review

Recent impressive performance improvements in computer architecture have not led to significant gains in the case of debugging. Software debugging often relies on inserting run-time software checks. In many cases, however, it is hard to find the root cause of a bug. Moreover, program execution typically slows down significantly, often by 10--100 times.To address this problem, this paper introduces the *intelligent watcher (iWatcher)*, a novel architectural scheme to monitor dynamic executio ...

Keywords: Architectural support, dynamic monitoring, software debugging, thread-level speculation (TLS)